Neural Networks 64 (2015) 49–58

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2015 Special Issue

Frame-by-frame language identification in short utterances using deep neural networks

Javier Gonzalez-Dominguez[a,b,*], Ignacio Lopez-Moreno[a], Pedro J. Moreno[a], Joaquin Gonzalez-Rodriguez[b]

[a] *Google Inc., NY, USA*

[b] *ATVS-Biometric Recognition Group, Universidad Autonoma de Madrid, Madrid, Spain*

ARTICLE INFO

Article history:
Available online 3 September 2014

Keywords:
DNNs
Real-time LID
i-vectors

ABSTRACT

This work addresses the use of deep neural networks (DNNs) in automatic language identification (LID) focused on short test utterances. Motivated by their recent success in acoustic modelling for speech recognition, we adapt DNNs to the problem of identifying the language in a given utterance from the short-term acoustic features. We show how DNNs are particularly suitable to perform LID in real-time applications, due to their capacity to emit a language identification posterior at each new frame of the test utterance. We then analyse different aspects of the system, such as the amount of required training data, the number of hidden layers, the relevance of contextual information and the effect of the test utterance duration. Finally, we propose several methods to combine frame-by-frame posteriors. Experiments are conducted on two different datasets: the public NIST Language Recognition Evaluation 2009 (3 s task) and a much larger corpus (of 5 million utterances) known as Google 5M LID, obtained from different Google Services. Reported results show relative improvements of DNNs versus the *i*-vector system of 40% in LRE09 3 second task and 76% in Google 5M LID.

1. Introduction

Automatic language identification (LID) refers to the process of automatically determining the language in a given speech sample (Muthusamy, Barnard, & Cole, 1994). The need for reliable LID is continuously growing due to several factors. Among them, the technological trend towards increased human interaction using hands-free, voice-operated devices and the need to facilitate the coexistence of a multiplicity of different languages in an increasingly globalized world.

In general, language discriminant information is spread across different structures or levels of the speech signal, ranging from low-level, short-term acoustic and spectral features to high-level, long-term features (i.e. phonotactic, prosodic). However, even though several high-level approaches are used as meaningful complementary sources of information (Ferrer, Scheffer, & Shriberg, 2010; Martinez, Lleida, Ortega, & Miguel, 2013; Zissman, 1996), most LID systems still include or rely on acoustic modelling (Gonzalez-Dominguez et al., 2010; Torres-Carrasquillo et al., 2010), mainly due to their better scalability and computational efficiency.

Indeed, computational cost plays an important role, as LID systems commonly act as a pre-processing stage for either machine systems (i.e. multilingual speech processing systems) or human listeners (i.e. call routing to a proper human operator) (Li, Ma, & Lee, 2013). Therefore, accurate and efficient behaviour in real-time applications is often essential, for example, when used for emergency call routing, where the response time of a fluent native operator is critical (Ambikairajah, Li, Wang, Yin, & Sethu, 2011; Muthusamy et al., 1994). In such situations, the use of high-level speech information may be prohibitive, as it often requires running one speech/phonetic recognizer per target language (Zissman & Berkling, 2001). Lightweight LID systems are especially necessary in cases where the application requires an implementation embedded in a portable device.

Driven by recent developments in speaker verification, the current state of the art in acoustic LID systems involves using *i*-vector front-end features followed by diverse classification mechanisms that compensate speaker and session variabilities (Brummer et al., 2012; Li et al., 2013; Sturim et al., 2011). The *i*-vector is a compact representation (typically from 400 to 600 dimensions) of a

* Corresponding author at: ATVS-Biometric Recognition Group, Universidad Autonoma de Madrid, Madrid, Spain. Tel.: +34 914977558.
E-mail address: javier.gonzalez@uam.es (J. Gonzalez-Dominguez).

http://dx.doi.org/10.1016/j.neunet.2014.08.006
0893-6080/© 2014 Elsevier Ltd. All rights reserved.

whole utterance, derived as a point estimate of the latent variables in a factor analysis model (Dehak, Torres-Carrasquillo, Reynolds, & Dehak, 2011; Kenny, Oullet, Dehak, Gupta, & Dumouchel, 2008). However, while proven to be successful in a variety of scenarios, *i*-vector-based approaches suffer from two major drawbacks when coping with real-time applications. First, the *i*-vector is a point estimate and its robustness quickly degrades as the amount of data used to derive it decreases. Note that the smaller the amount of data, the larger the variance of the posterior probability distribution of the latent variables, and thus, the larger the *i*-vector *uncertainty*. Second, in real-time applications, most of the costs associated with *i*-vector computation occur after completion of the utterance, which introduces an undesirable latency.

Motivated by the prominence of deep neural networks (DNNs), which surpass the performance of the previous dominant paradigm, Gaussian mixture models (GMMs), in diverse and challenging machine learning applications – including acoustic modelling (Hinton et al., 2012; Mohamed, Dahl, & Hinton, 2012), visual object recognition (Ciresan, Meier, Gambardella, & Schmidhuber, 2010), and many others (Yu & Deng, 2011) – we previously introduced a successful LID system based on DNNs in Lopez-Moreno et al. (2014). Unlike previous works on using shallow or convolutional neural networks for small LID tasks (Cole, Inouye, Muthusamy, & Gopalakrishnan, 1989; Leena, Srinivasa Rao, & Yegnanarayana, 2005; Montavon, 2009), this was, to the best of our knowledge, the first time that a DNN scheme was applied at a large scale for LID and benchmarked against alternative state-of-the-art approaches. Evaluated using two different datasets—the NIST LRE 2009 (3 s task) and Google 5M LID—this scheme significantly outperformed several *i*-vector-based state-of-the-art systems (Lopez-Moreno et al., 2014).

In the current study, we explore different aspects that affect DNN performance, with a special focus on very short utterances and real-time applications. We believe that the DNN-based system is a suitable candidate for this kind of application, as it could potentially generate decisions at each processed frame of the test speech segment, typically every 10 ms. Through this study, we assess the influence of several factors on the performance, namely: (a) the amount of required training data, (b) the topology of the network, (c) the importance of including the temporal context, and (d) the test utterance duration. We also propose several blind techniques to combine frame-by-frame posteriors obtained from the DNN to get identification decisions.

We conduct the experiments using the following LID datasets: a dataset built from Google data, hereafter, Google 5M LID corpus and the NIST Language Recognition Evaluation 2009 (LRE'09). First, by means of the Google 5M LID corpus, we evaluate the performance in a real application scenario. Second, we check if the same behaviour is observed in a familiar and standard evaluation framework for the LID community. In both cases, we focus on short test utterances (up to 3 s).

The rest of this paper is organized into the following sections. Section 2 defines a reference system based on *i*-vectors. The proposed DNN system is presented in Section 3. The experimental protocol and datasets are described in Section 4. Next, we examine the behaviour of our scheme over a range of configuration parameters in both the task and the neural network topology. Finally, Sections 6 and 7 are devoted to presenting the conclusions of the study and potential future work.

2. Baseline system: *i*-vector

Currently, most acoustic approaches to perform LID rely on *i*-vector technology (Dehak, Kenny, Dehak, Dumouchel, & Ouellet, 2011). All such approaches, while sharing *i*-vectors as a feature representation, differ in the type of classifier used to perform the final language identification (Martinez, Plchot, Burget, Glembek, & Matejka, 2011). In the rest of this section we describe: (a) the *i*-vector extraction procedure, (b) the *i*-vector classifier used in this study, and (c) the configuration details of our baseline *i*-vector system. This system will serve us as the baseline system.

2.1. I-vector extraction

Based on the MAP adaptation approach in a GMM framework (Reynolds, 1995), utterances in language or speaker recognition are typically represented by the accumulated zero- and centred first-order Baum–Welch statistics, N and F, respectively, computed from a Universal Background Model (UBM) λ. For the UBM mixture $m \in 1, \ldots, C$, with mean, μ_m, the corresponding zero- and centred first-order statistics are aggregated over all frames of the utterance as

$$N_m = \sum_t p(m|o_t, \lambda) \tag{1}$$

$$F_m = \sum_t p(m|o_t, \lambda)(o_t - \mu_m), \tag{2}$$

where $p(m|o_t, \lambda)$ is the Gaussian occupation probability for the mixture m given the spectral feature observation $o_t \in \Re^D$ at time t.

The total variability model, hereafter TV, can be seen as a classical FA generative model (Bishop, 2007), with observed variables given by the *supervector* (CD $\times$ 1) of stacked statistics $F = \{F_1, F_2, \ldots, F_C\}$. In the TV model, the vector of hidden variables $w \in \Re^L$ is known as the utterance *i*-vector. Observed and hidden variables are related by the rectangular low rank matrix $T \in \Re^{CD\times L}$

$$N^{-1}F = Tw, \tag{3}$$

where the zero-order statistics N are represented by a block diagonal matrix $\in \Re^{CD\times CD}$, with C diagonal $D \times D$ blocks. The mth component block is the matrix $N_m I_{(D\times D)}$. Given the imposed Gaussian distributions of $p(w)$ and $p(F|w)$, it can be seen that the mean of the posterior $p(w|F)$ is given by

$$w = (I + T^t \Sigma^{-1} NT)^{-1} T^t \Sigma^{-1} F, \tag{4}$$

where $\Sigma \in \Re^{CD\times CD}$ is the diagonal covariance matrix of F. The TV model is thus a data driven model with parameters $\{\lambda, T, \Sigma\}$. Kenny et al. (2008) provides a more detailed explanation of the derivation of these parameters, using the EM algorithm.

2.2. Classification

Since T constrains all the variabilities (i.e. language, speaker, session), and it is shared for all the language models/excerpts, the *i*-vectors, w, can be seen as a new input feature to classify. Further, several classifiers—either discriminative (i.e. Logistic Regression) or generative (i.e. the Gaussian classifier and linear discriminant analysis)—can be used to perform classification (Martinez et al., 2011). In this study, we utilized LDA, followed by cosine distance (LDA_CS), as the classifier.

Even though using a more sophisticated classifier (Lopez-Moreno et al., 2014) would have resulted in slightly increased performance, we chose the LDA_CS considering the trade-off between performance and computational time efficiency. In this framework, the similarity measure (score) of the two given *i*-vectors, w_1 and w_2, is obtained as

$$S_{w_1,w_2} = \frac{(A^t w_1)(A^t w_2)}{\sqrt{(A^t w_1)(A^t w_1)}\sqrt{(A^t w_2)(A^t w_2)}} \tag{5}$$

where A is the LDA matrix.

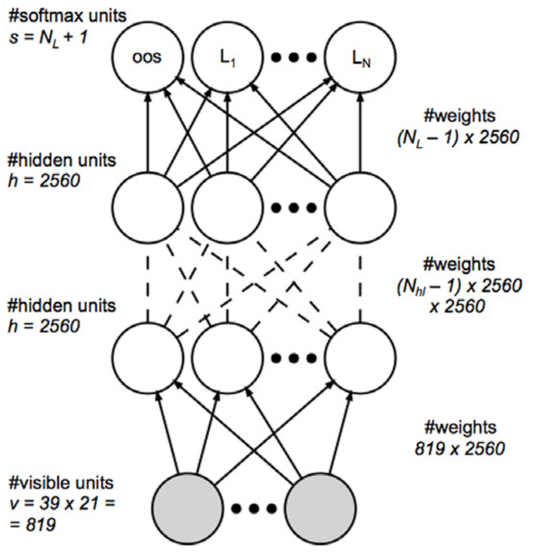


Fig. 1. DNN network topology.

2.3. Feature extraction and configuration parameters

As input features for this study we used perceptual linear predictive (PLP) coefficients (Hermansky, 1990). In particular, 13 PLP coefficients augmented with *delta* and *delta–delta* features (39 dimensions total) were extracted with a 10 ms frame rate over 25 ms long windows. From those features, we built a Universal Background Model of 1024 components. The total variability matrix was trained by using PCA and a posterior refinement of 10 EM iterations (Dehak, Kenny et al., 2011), keeping just the top 400 eigenvectors. We then derived the *i*-vectors using the standard methodology presented in Section 2.1. In addition, we filtered out silence frames by using an energy-based voice activity detector.

3. DNN as a language identification system

Recent findings in the field of speech recognition have shown that significant accuracy improvements over classical GMM schemes can be achieved through the use of deep neural networks, either to generate GMM features or to directly estimate acoustic model scores. Among the most important advantages of DNNs is their multilevel distributed representation of the input (Hinton et al., 2012). This fact makes the DNN an exponentially more compact model than GMMs. In addition DNNs do not require detailed assumptions about the input data distribution (Mohamed, Hinton, & Penn, 2012) and have proven successful in exploiting large amounts of data, reaching more robust models without lapsing into overtraining. All of these factors motivate the use of DNN in language identification. The rest of this section describes the architecture and the practical implementation of the DNN system.

3.1. Architecture

The DNN used in this work is a fully-connected feed-forward neural network with hidden units implemented as rectified linear units (ReLUs). Thus, an input at level j, x_j, is mapped to its corresponding activation y_j (input of the layer above) as

$$y_j = \mathrm{ReLU}(x_j) = \max(0, x_j) \tag{6}$$

$$x_j = b_j + \sum_i w_{ij} y_i \tag{7}$$

where i is an index over the units of the layer below and b_j is the bias of the unit j.

The output layer is then configured as a *softmax*, where hidden units map input x_j to a class probability p_j in the form

$$p_j = \frac{\exp(x_j)}{\sum_l \exp(x_l)} \tag{8}$$

where l is an index over all the classes.

As a cost function for backpropagating gradients in the training stage, we use the cross-entropy function defined as

$$C = -\sum_j t_j \log p_j \tag{9}$$

where t_j represents the target probability of the class j for the current evaluated example, taking a value of either 1 (true class) or 0 (false class).

3.2. Implementing DNN for language identification

From the conceptual architecture explained above, we built a language identification system to work at the frame level as follows.

As the input of the net we used the same features as the *i*-vector baseline system (39 PLP). Specifically, the input layer was fed with 21 frames formed by stacking the current processed frame and its ± 10 left/right neighbours. Thus, the input layer comprised a total number of 819 (21×39) visible units, v.

On top of the input layer, we stacked a total number of N_{hl} (8) hidden layers, each containing h (2560) units. Then, we added the softmax layer, whose dimension (s) corresponds to the number of target languages (N_L) plus one extra output for the out-of-set (*OOS*) languages. This OOS class, devoted to non-known test languages not seen in training time, could in future allow us to use the system in open-set identification scenarios. Overall, the net was defined by a total of w free parameters (weights + bias), $w = (v+1)h + (N_{hl}-1)(h+1)h + (h+1)s$ ($\sim$48M). The complete topology of the network is depicted in Fig. 1.

Regarding the training procedure, we used asynchronous stochastic gradient descent within the DistBelief framework (Dean et al., 2012), a software framework that uses computing clusters with thousands of machines to train large models. The learning rate and minibatch size were fixed to 0.001 and 200 samples.[1]

Note that the presented architecture works at the frame level, meaning that each single frame (plus its corresponding context) is fed-forward through the network, obtaining a class posterior probability for all of the target languages. This fact makes the DNNs particularly suitable for real-time applications since, unlike other approaches (i.e. *i*-vectors), we can potentially make a decision about the language at each new frame. Indeed, at each frame, we can combine the evidence from past frames to get a single similarity score between the test utterance and the target languages. A simple way of doing this combination is to assume that frames are independent and multiply the posterior estimates of the last layer. The score s_l for the language l of a given test utterance is computed by multiplying the output probabilities p_l obtained for all its frames, or equivalently, accumulating the logs as

$$s_l = \frac{1}{N} \sum_{t=1}^{N} \log p(L_l | x_t, \theta) \tag{10}$$

where $p(L_l|x_t, \theta)$ represents the class probability output for the language l corresponding to the input example at time t, x_t by using the DNN defined by parameters θ.

[1] We define sample as the input of the DNN: the feature representation of a single frame besides those from its adjacent frames forming the context.

Table 1
List of the Google 5M LID (above) and LRE'09 (below) languages considered.

Locale/Abbrev.	Language
Google 5M	
ar-EG	Arabic (Egypt)
ar-GULF	Arabic (Persian Gulf)
ar-LEVANT	Arabic (Levant)
bg-BG	Bulgarian
cs-CZ	Czech
de-DE	German
en-GB	English (United Kingdom)
en-IN	English (India)
en-US	English (USA)
en-ZA	English (South Africa)
es-419	Spanish (Latin America/Caribbean)
es-AR	Spanish (Argentina)
es-ES	Spanish (Spain)
fi-FI	Finish
fr-FR	French
he-IL	Hebrew (Israel)
hu-HU	Hungarian
id-ID	Indonesian
it-IT	Italian
ja-JP	Japanese
ko-KR	Korean (South Korea)
ms-MY	Malay
nl-NL	Dutch
pt-BR	Portuguese (Brazilian)
pt-PT	Portuguese (Portugal)
ro-RO	Romanian
ru-RU	Russian
sk-SK	Slovak
sr-RS	Serbian
sv-SE	Sweden
tr-TR	Turkish
zh-cmn-Hans-CN	Chinese (Mandarin)
zh-cmn-Hant-TW	Chinese (Taiwan)
zh-yue-hant-HK	Chinese (Cantonese)
LRE'09	
en	English (USA)
es	Spanish (Latin America/Caribbean)
fa	Farsi
fr	French
ps	Pashto
ru	Russian
ur	Urdu
zh	Chinese (Mandarin)

Table 2
Data description of the Google 5M LID and LRE09 subcorpus.

Database	$\#N_L$	Train (h)	Test (#files)	Test length (avg. on s)
Google 5M	34	2975	51 000	4.2
LRE09_VOA_3s	8	1600	2 916	3
LRE09_VOA_realtime	8	1600	11 276	(0.1 s–3 s)

4. Datasets and evaluation metrics

We conducted experiments on two different databases following the standard protocol provided by NIST in LRE 2009 (NIST, 2009). Particularly, we used the LRE'09 corpus and a corpus generated from Google Voice services. This followed a two-fold goal: first, to evaluate the proposed methods with a large collection of real application data, and second, to provide a benchmark comparable with other related works in the area by using the well-known LRE'09 framework.

4.1. Databases

4.1.1. Google 5M LID corpus

We generated the Google 5M LID corpus dataset by randomly picking anonymized queries from several Google speech recognition services such as Voice Search or the Speech Android API. Following the user's phone Voice Search language settings, we labelled a total of ∼5 million utterances, 150 k utterances by 34 different locales (25 languages + 9 dialects) yielding ∼87,5 h of speech per language and a total of ∼2975 h. A held-out test set of 1 k utterances per language was created while the remainder was used for training and development. Involved languages and data description are presented in Tables 1 and 2 respectively.

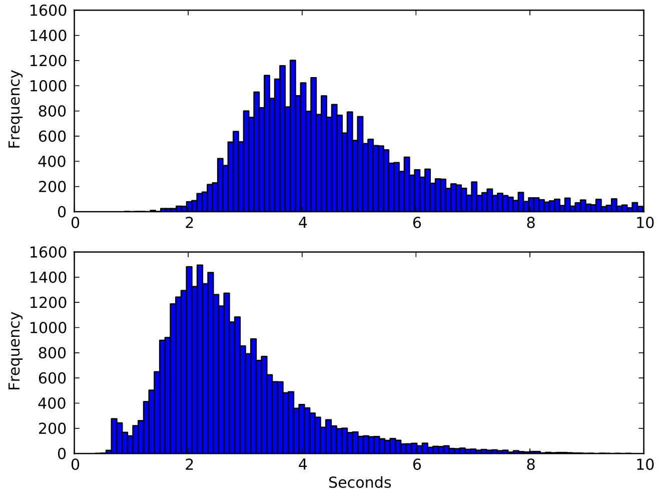


Fig. 2. Histograms of durations of the Google 5M LID test utterances. Original speech signals (above) and after voice activity detection (below).

An automatic speech recognition system was used to discard non-speech queries. Selected queries ranged from 1 up to 10 s in duration with average speech content of 2.1 s. Fig. 2 shows the duration distribution before and after doing this activity detection process.

Privacy issues do not allow Google to link the user identity with the spoken utterance and therefore, determining the exact number of speakers involved in this corpus is not possible. However, it is reasonable to consider that the total number of speakers is very large.

4.1.2. Language recognition evaluation 2009 dataset

The LRE evaluation in 2009 included, for the first time, data coming from two different audio sources. Besides Conversational Telephone Speech (CTS), used in the previous evaluations, telephone speech from broadcast news was used for both training and test purposes. Broadcast data were obtained via an automatic acquisition system from "Voice of America" news (VOA) where telephone and non-telephone speech are mixed.

Due to the large disparity on training material for every language (from ∼10 to ∼950 h), out of the 40 initial target languages (Liu, Zhang, & Hansen, 2012) we selected 8 representative languages for which up to 200 h of audio were available: US English (en), Spanish (es), Dari (fa), French (fr), Pashto (ps), Russian (ru), Urdu (ur), Chinese Mandarin (zh) (Table 1). Further, to avoid misleading result interpretation due to the unbalanced mix of CTS and VOA, all the data considered in this dataset were part of VOA.

As test material in LRE'09, we used a subset of the NIST LRE 2009 3 s condition evaluation set (as for training, we also discarded CTS test segments), yielding a total of 2916 test segments of the 8 selected languages. That makes a total of 23 328 trials. We refer this test dataset as LRE09_VOA_3s_test. For evaluating performance in real-time conditions, we used the VOA test segments for all the LRE'09 conditions (3 s, 10 s, 30 s) with at least 3 s of speech (according to our voice activity detector) that made a total of 11 276 files. Then we cut these recordings to build different duration subsets ranging from 0.1 to 3 s of speech. Specifically, we

Table 3
System performance (EER %) comparison per language on LRE09_VOA_3s_test. The *I*-vector baseline system vs. the DNN_8layers_200h system.

	Equal Error Rate (EER in %)								Avg.
	en	es	fa	fr	ps	ru	ur	zh	
Iv_200h	17.22	10.92	20.03	15.30	19.98	14.87	18.74	10.09	15.89
DNN_8layers_200h	**8.65**	**3.74**	**17.22**	**7.53**	**16.01**	**5.59**	**13.10**	**4.82**	**9.58**

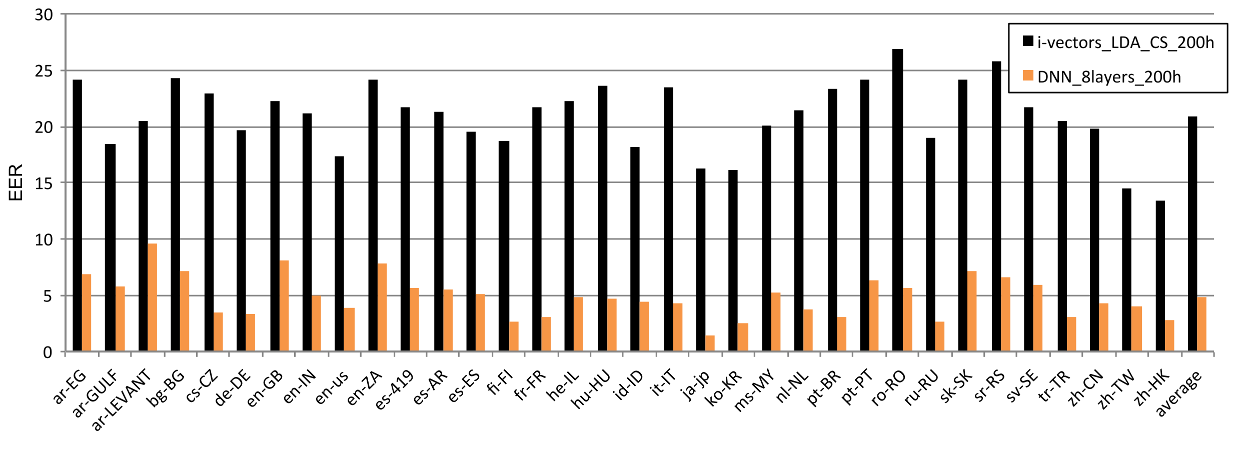


Fig. 3. System performance (EER %) comparison per language on Google 5M LID corpus. The *I*-vector baseline system vs. the DNN_8layers_200h system.

came up with 8 datasets of 11 276 files with durations: 0.1 s, 0.2 s, 0.5 s, 1 s, 1.5 s, 2.0 s, 2.5 s, and 3.0 s. We refer those test datasets as the LRE09_VOA_realtime_test benchmark.

4.2. Evaluation metrics

In order to assess the performance we used the accuracy and equal error rate (EER)[2] metrics. Language identification rates are measured in terms of accuracy, understanding this as the % of correctly identified trials when making hard decisions (by selecting the top scored language) language detection rates are measured in terms of per-language EER and for the sake of clarity we do not deal with the problem of setting optimal thresholds (calibration) as we previously did in Lopez-Moreno et al. (2014).

5. Experimental results

5.1. Global performance

We start our study by comparing the performance of the proposed DNN scheme with the baseline *i*-vector system on the LRE09_VOA_3s_test corpus. Table 3 summarizes this comparison in terms of EER. Results show how the DNN approach largely outperforms the *i*-vector system, obtaining up to a ∼40% relative improvement. An even larger improvement is obtained on the Google 5M corpus, where we found an average relative gain of ∼76% (see Fig. 3). Those results are especially remarkable since they are found on short test utterances and demonstrate the ability of the DNN to exploit discriminative information in large datasets.

It is also worth analysing the errors made by the DNN system as a function of the *similarity* of the different languages. We present in Fig. 7 the confusion matrix obtained using the DNN system on the Google 5M LID corpus. Confusion submatrices around dialects (i.e. ar-EG/ar-GULF/ar-LEVANT) illustrate the difficulty of dialect identification from spectral features in short utterances (Torres-Carrasquillo, Sturim, Reynolds, & McCree, 2008). These results suggest that exploiting just acoustic information might be not enough to reach accurate identification when dealing with dialects (Baker, Eddington, & Nay, 2009; Biadsy, 2011; Liu, Lei, & Hansen, 2010).

5.2. Number of hidden layers and training material

In this section, we evaluate two related aspects when training a DNN: the number of hidden layers and the amount of training material used. On the one hand, we want to exploit the ability showed by DNNs to improve the recognition performance while increasing the training, avoiding overfitting. On the other hand, we aim to get the lightest architecture possible without losing accuracy.

We started by fixing the available training material to its maximum in LRE'09 (200 h per language) and then reducing the number of hidden layers from 8 (DNN_8layers_200h) to 4 (DNN_4layers_200h) and 2 (DNN_2layers_200h). Table 4 summarizes those results. The net with 4 hidden layers seems to be more discriminative than the 2 hidden layers, and more interestingly, than the one with 8 hidden layers. In particular, on average, the DNN_4layers_200h outperforms by ∼8% in terms of EER the DNN_8layers_200h system, using half as many parameters.

As a further step, we swept the number of hours used per language from 1 to 200 h for the three nets. Fig. 4 shows the % accuracy as a function of the training hours per language. As expected, the bigger the amount of training data, the better the performance. However, the slope of this gain degrades when reaching 100 h per language. Indeed, from the 2 layer system, increasing the training material incurs in a minor degradation mostly due to underfitting. Again, it is clear from the results the need for a convenient tradeoff between the training data and number of parameters to optimize. In particular, our best configuration contains ∼21M parameters for ∼648M training samples.

5.3. Real-time identification

Taking now as reference the net with best performance so far (DNN_4_200h) we explored the performance degradation when limiting the test duration. The goal is to gain some insight about

[2] EER is the point on ROC or DET curve where false acceptance and true reject rates are equal.

Table 4
Effect of using different numbers of hidden layers. System performance (EER %) per language on LRE09_VOA_test_3s.

	Equal Error Rate (EER in %)								Avg.
	en	es	fa	fr	ps	ru	ur	zh	
DNN_2layers_200h	12.66	5.04	19.67	8.60	17.84	8.75	14.78	5.54	11.61
DNN_4layers_200h	**8.53**	**3.58**	**16.19**	**5.82**	**15.42**	6.38	**11.24**	**3.16**	**8.79**
DNN_8layers_200h	8.65	3.74	17.22	7.53	16.01	**5.59**	13.10	4.82	9.58

Table 5
Effect of using different left/right input contexts for the DNN_4layers_200h system. System performance (EER %) on the LRE09_VOA_realtime_test (3 s).

	Equal Error Rate (EER in %)								Avg.
	en	es	fa	fr	ps	ru	ur	zh	
No context	19.07	9.65	24.82	13.17	21.64	14.28	19.39	12.38	16.80
±10	8.42	**3.62**	15.89	**5.46**	14.54	6.31	**10.05**	**3.47**	8.47
±20	**7.71**	3.88	**15.49**	6.11	**12.90**	**6.09**	10.50	4.00	**8.33**
±30	9.44	4.53	16.24	7.95	14.40	7.96	12.07	5.23	9.72
±40	12.05	5.08	17.41	9.71	15.47	9.14	13.10	6.27	11.03
±50	9.85	5.71	19.26	8.80	14.54	7.76	13.37	6.51	10.72

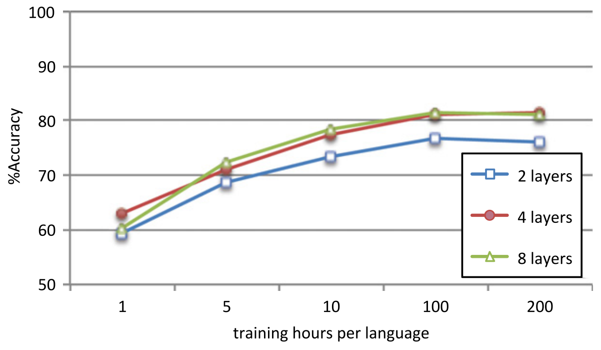


Fig. 4. DNN system performance (% accuracy) in function of the training time per language and the number of hidden layers. Results on LRE09_VOA_3s_test.

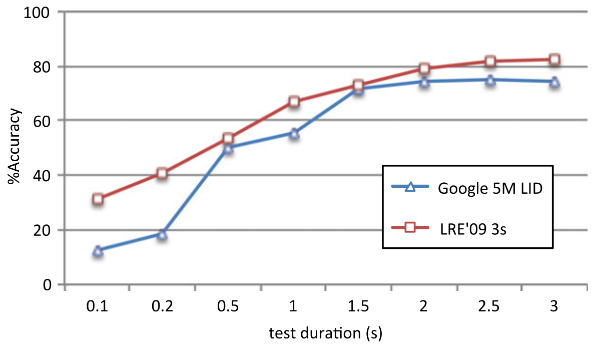


Fig. 5. DNN_4layers_200h system performance (% accuracy) in function of the test utterance duration. Results on LRE09_VOA_realtime_3s.

how long a test utterance must be to consider the identification *accurate*, a main concern in real-time applications.

Fig. 5 shows the average accuracy as a function of the test durations, for both test corpus LRE09_VOA_realtime_test and Google 5M LID. We highlight here two main points. Notice first that up to 0.5 s of speech (according to our voice activity detection) the identification accuracy is very poor (rates under 50% accuracy). Very quick decisions can lead systems to a bad user experience in real-time applications. Second, as expected, the larger the test duration, the better the performance. However, this practically saturates after 2 s. This suggests that a decision can be taken at this point without significant loss of accuracy even when we increase the number of target languages from 9 to 34.

A more detailed analysis per language can be seen in Table 7 for all the 34 languages involved in the Google 5M LID corpus, where we show that the previous conclusion holds true also for each individual language. Confusion matrices on the LRE09_VOA_realtime_test are also collected in Figs. A.8–A.11.

5.4. Temporal context

So far we have been using a fixed right/left context of ±10 frames respectively. That is, the input of our network, as mentioned in Section 3, is formed by stacking the features of every frame with its corresponding 10 to the left and 10 to the right neighbours. We explore in this section the effect of including a shorter/wider context for language identification.

The motivation behind using temporal information from a large number of frames lies in the idea of incorporating additional high-level information (i.e. phonetic, phonotactic and prosodic information). This idea has been largely and successfully exploited in language identification by using long-term phonotactic/prosodic tokenizations (Ferrer et al., 2010; Reynolds et al., 2003) or, in acoustic approaches, by using shifted-delta-cepstral features (Torres-Carrasquillo, Singer, Kohler, & Deller, 2002).

We modify the input of the network by stacking each frame with a symmetric context that ranges from 0 to 50 left and right neighbour frames; that is, we sweep from a context-free scheme to a maximum context that spans 0.5 s to the left and 0.5 s to the right (a total of 1 s context).

Table 5 summarizes the obtained results on the LRE09_VOA_realtime_test (3 s subcorpus) using the DNN_4_200h network. The importance of the context is apparent from first two rows. We observe a relative improvement of ∼49% from the ±10 context scheme with respect to the context-free one. We find the lowest EER when using ±20 frames of context. After this value the EER increases. This behaviour can be explained by understanding that as we demand our net to learn more 'high-level' rich features, we are also increasing the size of the input, therefore forcing the net to learn more complex features from the same amount of data. Fig. 6 collectsthe top 10 filters for a given minibatch (those which produce highest activations in the minibatch) extracted from the first hidden layer for the DNN_4_200h network. The distribution of those weights evidences how the DNN is using the context information.

Although the number of parameters of the input layer is affected by the size of the contextual window, the input layer represents less than the 25% of the model size. Thus, it seems that DNNs can lead to better modelling of the contextual information than competing approaches, such as GMM-based systems, which are traditionally more affected by the *curse of dimensionality*. Note that

Table 6
Comparison of different frame combination schemes for the DNN_4layers_200h. System performance (EER %) per language. Results on LRE09_VOA_3s_test.

	Equal Error Rate (EER in %)								Avg.
	en	es	fa	fr	ps	ru	ur	zh	
Product	**8.53**	**3.58**	**16.19**	**5.82**	**15.42**	6.38	**11.24**	**3.16**	**8.79**
Voting	12.66	5.04	19.67	8.60	17.84	8.75	14.78	5.54	11.61
Entropy	8.65	3.74	17.22	7.53	16.01	**5.59**	13.10	4.82	9.58

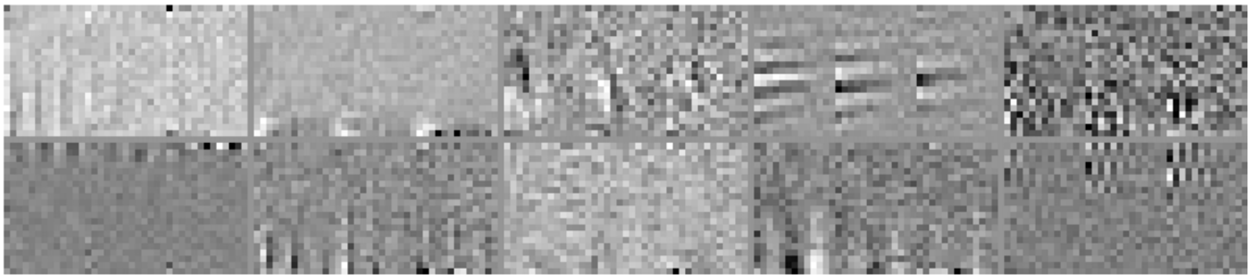
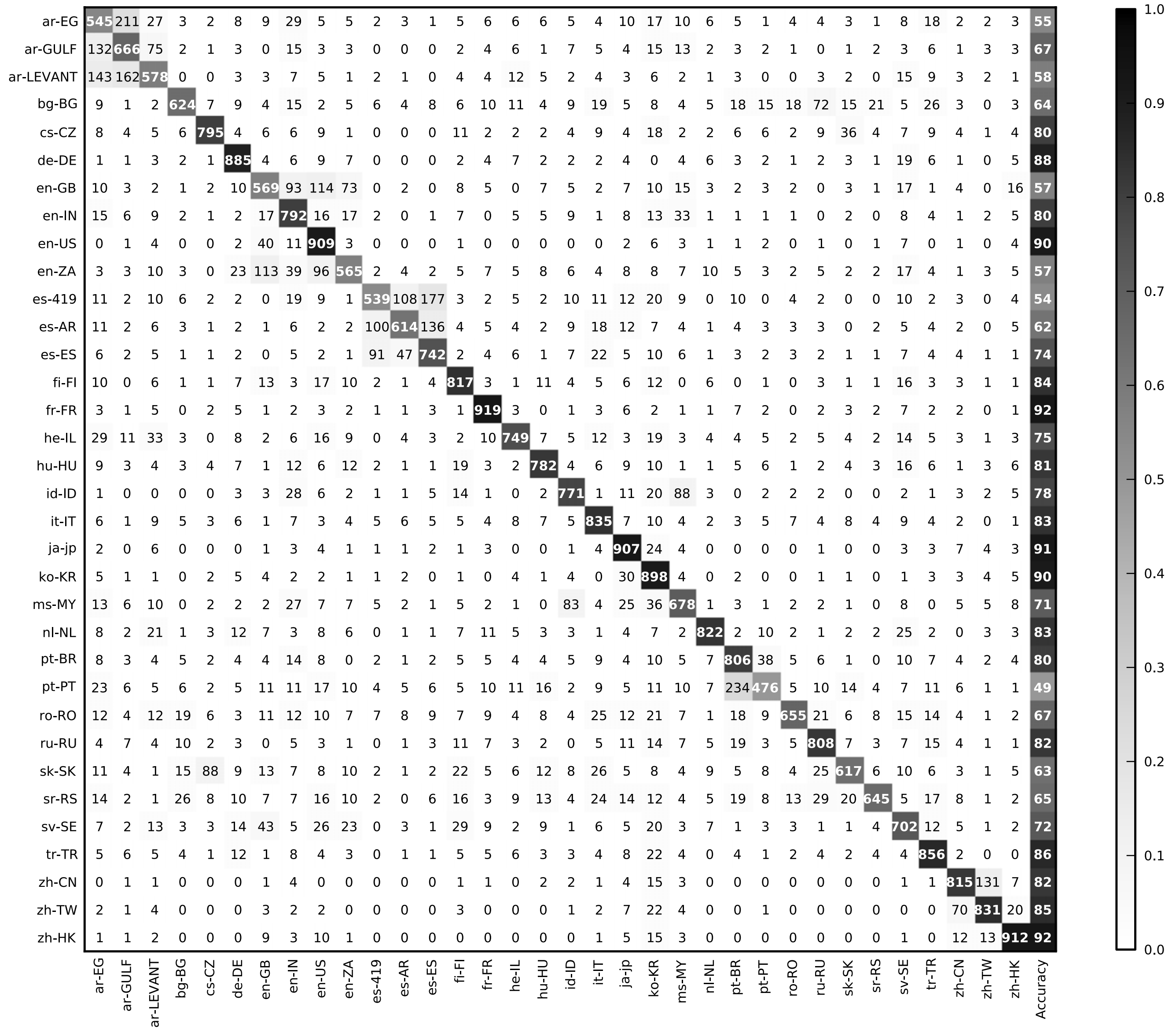


Fig. 6. Visualization of top 10 filters (those which produce highest activations in the given minibatch) of the first hidden layer using a ±10 context. Each filter is composed of 21 rows (number of frames stacked as input) and 39 columns (feature dimension).



	ar-EG	ar-GULF	ar-LEVANT	bg-BG	cs-CZ	de-DE	en-GB	en-IN	en-US	en-ZA	es-419	es-AR	es-ES	fi-FI	fr-FR	he-IL	hu-HU	id-ID	it-IT	ja-jp	ko-KR	ms-MY	nl-NL	pt-BR	pt-PT	ro-RO	ru-RU	sk-SK	sr-RS	sv-SE	tr-TR	zh-CN	zh-TW	zh-HK	Accuracy
ar-EG	545	211	27	3	2	8	9	29	5	5	2	3	1	5	6	6	6	5	4	10	17	10	6	5	1	4	4	3	1	8	18	2	2	3	55
ar-GULF	132	666	75	2	1	3	0	15	3	3	0	0	0	2	4	6	1	7	5	4	15	13	2	3	2	1	0	1	2	3	6	1	3	3	67
ar-LEVANT	143	162	578	0	0	3	3	7	5	1	2	1	0	4	4	12	5	2	4	3	6	2	1	3	0	0	3	2	0	15	9	3	2	1	58
bg-BG	9	1	2	624	7	9	4	15	2	5	6	4	8	6	10	11	4	9	19	5	8	4	5	18	15	18	72	15	21	5	26	3	0	3	64
cs-CZ	8	4	5	6	795	4	6	6	9	1	0	0	0	11	2	2	2	4	9	4	18	2	2	6	6	2	9	36	4	7	9	4	1	4	80
de-DE	1	1	3	2	1	885	4	6	9	7	0	0	0	2	4	7	2	2	2	4	0	4	6	3	2	1	2	3	1	19	6	1	0	5	88
en-GB	10	3	2	1	2	10	569	93	114	73	0	2	0	8	5	0	7	5	2	7	10	15	3	2	3	2	0	3	1	17	1	4	0	16	57
en-IN	15	6	9	2	1	2	17	792	16	17	2	0	1	7	0	5	5	9	1	8	13	33	1	1	1	1	0	2	0	8	4	1	2	5	80
en-US	0	1	4	0	0	2	40	11	909	3	0	0	0	1	0	0	0	0	0	2	6	3	1	1	2	0	1	0	1	7	0	1	0	4	90
en-ZA	3	3	10	3	0	23	113	39	96	565	2	4	2	5	7	5	8	6	4	8	8	7	10	5	3	2	5	2	2	17	4	1	3	5	57
es-419	11	2	10	6	2	2	0	19	9	1	539	108	177	3	2	5	2	10	11	12	20	9	0	10	0	4	2	0	0	10	2	3	0	4	54
es-AR	11	2	6	3	1	2	1	6	2	2	100	614	136	4	5	4	2	9	18	12	7	4	1	4	3	3	3	0	2	5	4	2	0	5	62
es-ES	6	2	5	1	1	2	0	5	2	1	91	47	742	2	4	6	1	7	22	5	10	6	1	3	2	3	2	1	1	7	4	4	1	1	74
fi-FI	10	0	6	1	1	7	13	3	17	10	2	1	4	817	3	1	11	4	5	6	12	0	6	0	1	0	3	1	1	16	3	3	1	1	84
fr-FR	3	1	5	0	2	5	1	2	3	2	1	1	3	1	919	3	0	1	3	6	2	1	1	7	2	0	2	3	2	7	2	2	0	1	92
he-IL	29	11	33	3	0	8	2	6	16	9	0	4	3	2	10	749	7	5	12	3	19	3	4	4	5	2	5	4	2	14	5	3	1	3	75
hu-HU	9	3	4	3	4	7	1	12	6	12	2	1	1	19	3	2	782	4	6	9	10	1	1	5	6	1	2	4	3	16	6	1	3	6	81
id-ID	1	0	0	0	0	3	3	28	6	2	1	1	5	14	1	0	2	771	1	11	20	88	3	0	2	2	2	0	0	2	1	3	2	5	78
it-IT	6	1	9	5	3	6	1	7	3	4	5	6	5	5	4	8	7	5	835	7	10	4	2	3	5	7	4	8	4	9	4	2	0	1	83
ja-jp	2	0	6	0	0	0	1	3	4	1	1	1	2	1	3	0	0	1	4	907	24	4	0	0	0	0	1	0	0	3	3	7	4	3	91
ko-KR	5	1	1	0	2	5	4	2	2	1	1	2	0	1	0	4	1	4	0	30	898	4	0	2	0	0	1	1	0	1	3	3	4	5	90
ms-MY	13	6	10	0	2	2	2	27	7	7	5	2	1	5	2	1	0	83	4	25	36	678	1	3	1	2	2	1	0	8	0	5	5	8	71
nl-NL	8	2	21	1	3	12	7	3	8	6	0	1	1	7	11	5	3	3	1	4	7	2	822	2	10	2	1	2	2	25	2	0	3	3	83
pt-BR	8	3	4	5	2	4	4	14	8	0	2	1	2	5	5	4	4	5	9	4	10	5	7	806	38	5	6	1	0	10	7	4	2	4	80
pt-PT	23	6	5	6	2	5	11	11	17	10	4	5	6	5	10	11	16	2	9	5	11	10	7	234	476	5	10	14	4	7	11	6	1	1	49
ro-RO	12	4	12	19	6	3	11	12	10	7	7	8	9	7	9	4	8	4	25	12	21	7	1	18	9	655	21	6	8	15	14	4	1	2	67
ru-RU	4	7	4	10	2	3	0	5	3	1	0	1	3	11	7	3	2	0	5	11	14	7	5	19	3	5	808	7	3	7	15	4	1	1	82
sk-SK	11	4	1	15	88	9	13	7	8	10	2	1	2	22	5	6	12	8	26	5	8	4	9	5	8	4	25	617	6	10	6	3	1	5	63
sr-RS	14	2	1	26	8	10	7	7	16	10	2	0	6	16	3	9	13	4	24	14	12	4	5	19	8	13	29	20	645	5	17	8	1	2	65
sv-SE	7	2	13	3	3	14	43	5	26	23	0	3	1	29	9	2	9	1	6	5	20	3	7	1	3	3	1	1	4	702	12	5	1	2	72
tr-TR	5	6	5	4	1	12	1	8	4	3	0	1	1	5	5	6	3	3	4	8	22	4	0	4	1	2	4	2	4	4	856	2	0	0	86
zh-CN	0	1	1	0	0	0	1	4	0	0	0	0	0	1	1	0	2	2	1	4	15	3	0	0	0	0	0	0	0	1	1	815	131	7	82
zh-TW	2	1	4	0	0	0	3	2	2	0	0	0	0	3	0	0	0	1	2	7	22	4	0	0	1	0	0	0	0	0	0	70	831	20	85
zh-HK	1	1	2	0	0	0	9	3	10	1	0	0	0	0	0	0	0	0	1	5	15	3	0	0	0	0	0	0	0	1	0	12	13	912	92

Fig. 7. Confusion matrix obtained by evaluating the DNN_8_200h system on the Google 5M LID corpus.

Table 7
System performance (accuracy %) by language and test utterance duration on Google 5M Database.

% Accuracy		Test utterance duration (s)							
Locale	Language	0.1	0.2	0.5	1.0	1.5	2.0	2.5	3.0
ar-EG	Arabic (Egypt)	12	14	38	41	52	56	**58**	57
ar-GULF	Arabic (Persian)	19	22	46	53	61	64	**69**	68
ar-LEVANT	Arabic (Levant)	43	51	54	48	**65**	64	61	62
bg-BG	Bulgarian	7	11	38	51	65	64	68	**72**
cs-CZ	Czech	2	6	45	69	71	75	79	**81**
de-DE	German	18	27	62	72	84	88	87	**89**
en-GB	English (United)	4	12	31	39	49	**54**	**54**	**54**
en-IN	English (India)	27	30	56	63	73	74	76	**78**
en-US	English (USA)	22	29	62	70	85	87	89	**91**
en-ZA	English (South)	4	7	34	45	46	51	56	**57**
es-419	Spanish (Latin)	6	8	25	41	47	50	52	**55**
es-AR	Spanish (Argentina)	6	8	35	50	53	56	58	**61**
es-ES	Spanish (Spain)	5	9	48	54	67	70	72	**73**
fi-FI	Finish	14	23	55	75	76	80	**82**	**82**
fr-FR	French	14	25	69	83	90	93	94	**95**
he-IL	Hebrew (Israel)	4	10	46	60	60	67	68	**70**
hu-HU	Hungarian	8	16	48	71	72	80	**82**	**82**
id-ID	Indonesian	13	21	45	62	69	72	75	**76**
it-IT	Italian	8	13	42	58	75	78	80	**81**
ja-JP	Japanese	18	25	68	87	89	91	94	**95**
ko-KR	Korean (South)	16	25	68	91	89	91	**92**	**92**
ms-MY	Malay	17	25	44	59	63	70	**72**	**72**
nl-NL	Dutch	6	12	56	68	76	80	80	**81**
pt-BR	Portuguese (Brazilian)	11	18	47	74	74	78	80	**81**
pt-PT	Portuguese (Portugal)	6	8	28	53	42	43	48	**49**
ro-RO	Romanian	7	12	34	43	56	61	64	**66**
ru-RU	Russian	5	11	52	70	83	**85**	**85**	**85**
sk-SK	Slovak	10	13	30	40	48	51	55	**58**
sr-RS	Serbian	6	9	35	54	55	59	60	**62**
sv-SE	Sweden	10	16	42	62	65	70	**73**	71
tr-TR	Turkish	5	10	55	78	79	82	83	**85**
zh-cmn-Hans-CN	Chinese (Mandarin)	12	16	54	76	75	76	80	**82**
zh-cmn-Hant-TW	Chinese (Taiwan)	12	22	63	78	80	83	83	**85**
zh-yue-hant-HK	Chinese (Cantonese)	15	25	68	81	88	91	90	**91**

the relative gains reported in this analysis (∼50%) surpass previous attempts reported in the literature in including contextual information using the GMM paradigm (Torres-Carrasquillo et al., 2002). We refer also to Li and Narayanan (2014) for a extensive comparison of different features in language identification over an *i*-vector-based framework.

5.5. *Frame-by-frame posteriors combination*

One of the features that make DNNs particularly suitable for real-time applications is their ability to generate frame-by-frame posteriors. Indeed we can derive decisions about the language identification at each frame. Here we aim to study how we can combine frame posteriors into a single utterance-level score.

Probably the most standard way to perform this combination is assuming frame independence and using the product rule (see Section 3). That is, simply compute the product of the posteriors frame-by-frame as the new single score vector. Another common and simple approach used in the literature is plurality voting, where, at each frame, the language associated with the highest posterior receives a single *vote* while the rest receive none. The voting scheme aims to control the negative effect of outlier scores. The score for a given language l, s_l, is then computed by counting the received votes over all the frames as

$$s_l = \sum_{t=1}^{N} \delta(p(L_l|x_t,\theta)), \tag{11}$$

with δ function defined as

$$\delta(p(L_l|x_t,\theta)) \begin{cases} 1, & \text{if } l = \arg\max_{l}(p(L_l|x_t,\theta)) \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

A more interesting approach, among *blind* techniques (no need for training), is to weight the posteriors of every frame as a function of the entropy of its posterior distribution. The idea here is to penalize those frames whose distribution of posteriors across the set of languages tends to be uniform (high entropy). This approach was successfully applied in Misra, Bourlard, and Tyagi (2003), resulting in a performance improvement when working with mismatched training and test datasets. The resulting score for language l, s_l, is computed as

$$s_l = \sum_{t=1}^{N} \log\left(\frac{1}{h_t} p(L_l|x_t,\theta)\right) \tag{13}$$

where the weight for frame t is the inverse of its entropy

$$h_t = -\sum_{l=1}^{N} p(L_l|x_t,\theta) \log_2 p(L_l|x_t,\theta). \tag{14}$$

Table 6 compares these three different combination schemes: product, voting and entropy on the LRE09_VOA_3s_test corpus. Results show a better performance of the simple product rule compared to the other approaches, with voting the worst choice. This result suggests that making binary decisions at a frame level leads to a performance degradation. Although the entropy scheme does not help in this scenario, it should be considered when working with more noisy environments.

6. Conclusion

In this work, we present a detailed analysis of the use of deep neural networks (DNNs) for automatic language identification (LID) of short utterances. Guided by the success of DNNs for acoustic modelling in speech recognition, we explore the capacity of DNNs to learn language information embedded in speech signals.

Through this study, we also explore the limits of the proposed scheme for real-time applications, evaluating the accuracy of the system when using very short test utterances ($\leq$3 s). We find, for our proposed DNN scheme, that while more than 0.5 s is needed to obtain over 50% accuracy rates, 2 s are enough to reach accuracy rates of over 90%. Further, we experiment with the amount of training material, the number of hidden layers and the combination of frame posteriors. We also analyse the relevance of including the temporal context, which is critical to achieving high performance in LID.

Results using NIST LRE 2009 (8 languages selected) and Google 5M LID datasets (25 languages + 9 dialects) demonstrate that DNNs outperform current state-of-art *i*-vector-based approaches when dealing with short test durations. Finally, we demonstrated that using a frame-by-frame approach, DNNs can be successful applied for real-time applications.

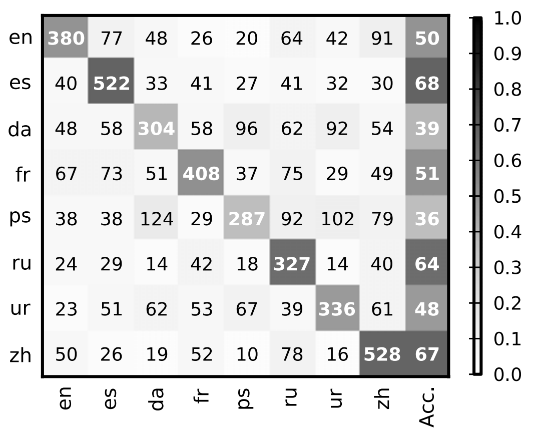
7. Future work

We identified several areas where further investigation is needed. Among them, establishing a more appropriate combination of frame posteriors obtained in DNNs; exploring different fusions among DNNs and *i*-vector systems (Saon, Soltau, Nahamoo, & Picheny, 2013); and dealing with unbalanced training data. Note that even though we proposed different ways of combining posteriors, all of them are blind techniques (no need for training). This fact is due to we focused on real-time applications and simple approaches. However, other data-driven methods could be more appropriate for combining posteriors.

Further other neural network architectures should also be explored. For instance, recurrent neural networks might be an elegant solution to incorporate contextual information. Also, convolutional neural networks might help to reduce the number of parameters of our model.

Another promising approach is the use of the activations of the last hidden layer as bottleneck features. Then, *i*-vector-based systems or another classification architecture could be trained over those bottleneck features, rather than over classical features, such as PLP or MFCC.

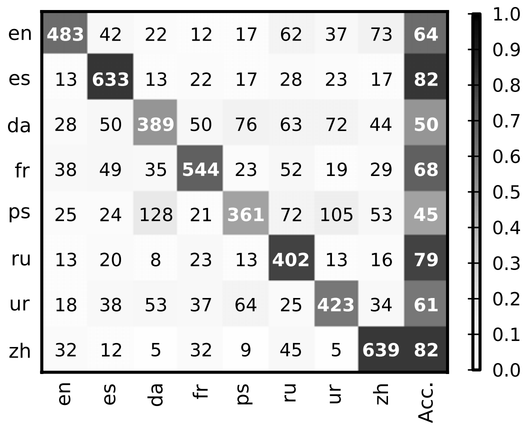
Appendix. Extended results



	en	es	da	fr	ps	ru	ur	zh	Acc.
en	380	77	48	26	20	64	42	91	50
es	40	522	33	41	27	41	32	30	68
da	48	58	304	58	96	62	92	54	39
fr	67	73	51	408	37	75	29	49	51
ps	38	38	124	29	287	92	102	79	36
ru	24	29	14	42	18	327	14	40	64
ur	23	51	62	53	67	39	336	61	48
zh	50	26	19	52	10	78	16	528	67

1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

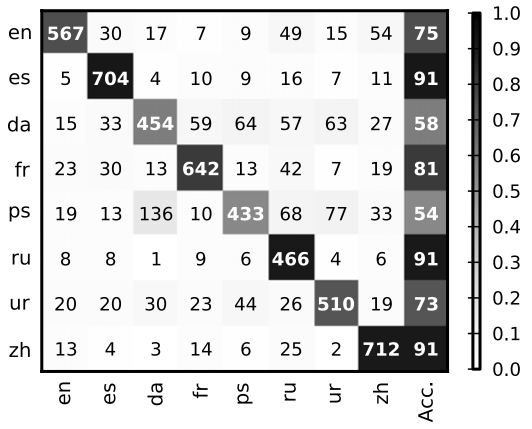
Fig. A.8. DNN_4layers_200h confusion matrix on LRE'09 (0.5 s test).



	en	es	da	fr	ps	ru	ur	zh	Acc.
en	483	42	22	12	17	62	37	73	64
es	13	633	13	22	17	28	23	17	82
da	28	50	389	50	76	63	72	44	50
fr	38	49	35	544	23	52	19	29	68
ps	25	24	128	21	361	72	105	53	45
ru	13	20	8	23	13	402	13	16	79
ur	18	38	53	37	64	25	423	34	61
zh	32	12	5	32	9	45	5	639	82

1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

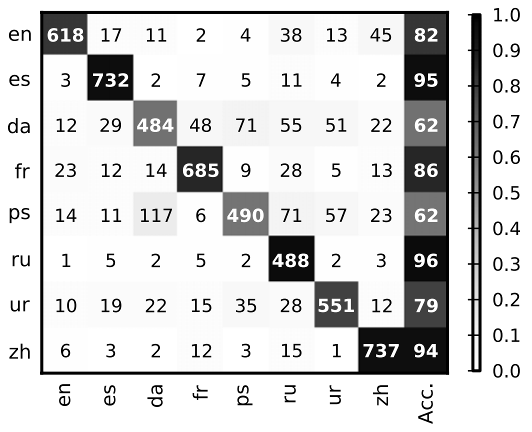
Fig. A.9. DNN_4layers_200h confusion matrix on LRE'09 (1 s test).



	en	es	da	fr	ps	ru	ur	zh	Acc.
en	567	30	17	7	9	49	15	54	75
es	5	704	4	10	9	16	7	11	91
da	15	33	454	59	64	57	63	27	58
fr	23	30	13	642	13	42	7	19	81
ps	19	13	136	10	433	68	77	33	54
ru	8	8	1	9	6	466	4	6	91
ur	20	20	30	23	44	26	510	19	73
zh	13	4	3	14	6	25	2	712	91

1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

Fig. A.10. DNN_4layers_200h confusion matrix on LRE'09 (2 s test).



	en	es	da	fr	ps	ru	ur	zh	Acc.
en	618	17	11	2	4	38	13	45	82
es	3	732	2	7	5	11	4	2	95
da	12	29	484	48	71	55	51	22	62
fr	23	12	14	685	9	28	5	13	86
ps	14	11	117	6	490	71	57	23	62
ru	1	5	2	5	2	488	2	3	96
ur	10	19	22	15	35	28	551	12	79
zh	6	3	2	12	3	15	1	737	94

1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

Fig. A.11. DNN_4layers_200h confusion matrix on LRE'09 (3 s test).

References

Ambikairajah, E., Li, H., Wang, L., Yin, B., & Sethu, V. (2011). Language identification: A tutorial. *IEEE Circuits and Systems Magazine*, *11*(2), 82–108. http://dx.doi.org/10.1109/MCAS.2011.941081.

Baker, W., Eddington, D., & Nay, L. (2009). Dialect identification: The effects of region of origin and amount of experience. *American Speech*, *84*(1), 48–71.

Biadsy, F. (2011). *Automatic dialect and accent recognition and its application to speech recognition*. (Ph.D. thesis). Columbia University.

Bishop, C. (2007). *Information science and statistics, Pattern recognition and machine learning* (1st ed.). Springer.

Brummer, N., Cumani, S., Glembek, O., Karafiát, M., Matejka, P., Pesan, J., et al. (2012). Description and analysis of the Brno276 system for LRE2011. In *Proceedings of Odyssey 2012: The speaker and language recognition workshop* (pp. 216–223). International Speech Communication Association.

Ciresan, D., Meier, U., Gambardella, L., & Schmidhuber, J. (2010). Deep big simple neural nets excel on handwritten digit recognition, CoRR abs/1003.0358.

Cole, R., Inouye, J., Muthusamy, Y., & Gopalakrishnan, M. (1989). Language identification with neural networks: A feasibility study. In *IEEE Pacific Rim conference on communications, computers and signal processing, 1989. Conference proceeding.* (pp. 525–529) http://dx.doi.org/10.1109/PACRIM.1989.48417.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q., et al. (2012). Large scale distributed deep networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1232–1240).

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., & Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, *19*(4), 788–798.

Dehak, N., Torres-Carrasquillo, P. A., Reynolds, D. A., & Dehak, R. (2011). Language recognition via *i*-vectors and dimensionality reduction. In *INTERSPEECH* (pp. 857–860). ISCA.

Ferrer, L., Scheffer, N., & Shriberg, E. (2010). A comparison of approaches for modeling prosodic features in speaker recognition. In *International conference on acoustics, speech, and signal processing* (pp. 4414–4417) http://dx.doi.org/10.1109/ICASSP.2010.5495632.

Gonzalez-Dominguez, J., Lopez-Moreno, I., Franco-Pedroso, J., Ramos, D., Toledano, D., & Gonzalez-Rodriguez, J. (2010). Multilevel and session variability compensated language recognition: ATVS-UAM systems at NIST LRE 2009. *IEEE Journal of Selected Topics in Signal Processing*, *4*(6), 1084–1093. http://dx.doi.org/10.1109/JSTSP.2010.2076071.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, *87*(4), 1738–1752.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97. http://dx.doi.org/10.1109/MSP.2012.2205597.

Kenny, P., Oullet, P., Dehak, V., Gupta, N., & Dumouchel, P. (2008). A study of interspeaker variability in speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, *16*(5), 980–988.

Leena, M., Srinivasa Rao, K., & Yegnanarayana, B. (2005). Neural network classifiers for language identification using phonotactic and prosodic features. In *Proceedings of 2005 international conference on intelligent sensing and information processing, 2005* (pp. 404–408) http://dx.doi.org/10.1109/ICISIP.2005.1529486.

Li, H., Ma, B., & Lee, K. A. (2013). Spoken language recognition: From fundamentals to practice. *Proceedings of the IEEE*, *101*(5), 1136–1159. http://dx.doi.org/10.1109/JPROC.2012.2237151.

Li, M., & Narayanan, S. (2014). Simplified supervised *i*-vector modeling with application to robust and efficient language identification and speaker verification, Computer Speech and Language.

Liu, G., Lei, Y., & Hansen, J. H. (2010). Dialect identification: Impact of difference between read versus spontaneous speech. In *EUSIPCO-2010* (pp. 2003–2006).

Liu, G., Zhang, C., & Hansen, J. H. L. (2012). A linguistic data acquisition front-end for language recognition evaluation. In *Proc. Odyssey, Singapore*.

Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., & Moreno, P. (2014). Automatic language identification using deep neural networks. In *IEEE International conference on acoustics, speech, and signal processing*.

Martinez, D., Lleida, E., Ortega, A., & Miguel, A. (2013). Prosodic features and formant modeling for an ivector-based language recognition system. In *2013 IEEE International conference on acoustics, speech and signal processing, ICASSP* (pp. 6847–6851) http://dx.doi.org/10.1109/ICASSP.2013.6638988.

Martinez, D., Plchot, O., Burget, L., Glembek, O., & Matejka, P. (2011). Language recognition in ivectors space. In *INTERSPEECH* (pp. 861–864). ISCA.

Misra, H., Bourlard, H., & Tyagi, V. (2003). New entropy-based combination rules in HMM/ANN multi-stream ASR. In *2003 IEEE International conference on acoustics, speech, and signal processing, 2003. Proceedings, ICASSP'03, Vol. 2* (pp. II-741-4) http://dx.doi.org/10.1109/ICASSP.2003.1202473.

Mohamed, A., Dahl, G., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Language Processing*, *20*(1), 14–22. http://dx.doi.org/10.1109/TASL.2011.2109382.

Mohamed, A. Rahman, Hinton, G. E., & Penn, G. (2012). Understanding how deep belief networks perform acoustic modelling. In *ICASSP* (pp. 4273–4276). IEEE.

Montavon, G. (2009). Deep learning for spoken language identification. In *NIPS Workshop on deep learning for speech recognition and related applications*.

Muthusamy, Y., Barnard, E., & Cole, R. (1994). Reviewing automatic language identification. *IEEE Signal Processing Magazine*, *11*(4), 33–41. http://dx.doi.org/10.1109/79.317925.

NIST, 2009. The 2009 NIST SLR Evaluation Plan. www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf.

Reynolds, D. (1995). Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, *17*(1–2), 91–108.

Reynolds, D., Andrews, W., Campbell, J., Navratil, J., Peskin, B., & Adami, A. et al. (2003). The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition. In *IEEE International conference on acoustics, speech, and signal processing, Vol. 4* (pp. 784–787).

Saon, G., Soltau, H., Nahamoo, D., & Picheny, M. (2013). Speaker adaptation of neural network acoustic models using *i*-vectors. In *2013 IEEE Workshop on automatic speech recognition and understanding, ASRU* (pp. 55–59) http://dx.doi.org/10.1109/ASRU.2013.6707705.

Sturim, D., Campbell, W., Dehak, N., Karam, Z., McCree, A., & Reynolds, D. et al. (2011). The MIT LL 2010 speaker recognition evaluation system: Scalable language-independent speaker recognition. In *2011 IEEE International conference on acoustics, speech and signal processing, ICASSP* (pp. 5272–5275) http://dx.doi.org/10.1109/ICASSP.2011.5947547.

Torres-Carrasquillo, P., Singer, E., Gleason, T., McCree, A., Reynolds, D., & Richardson, F. et al. (2010). The MITLL NIST LRE 2009 language recognition system. In *2010 IEEE International conference on acoustics speech and signal processing, ICASSP* (pp. 4994–4997) http://dx.doi.org/10.1109/ICASSP.2010.5495080.

Torres-Carrasquillo, P. A., Singer, E., Kohler, M. A., & Deller, J. R. (2002). Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. In *ICSLP, Vol. 1* (pp. 89–92).

Torres-Carrasquillo, P. A., Sturim, D. E., Reynolds, D. A., & McCree, A. (2008). Eigen-channel compensation and discriminatively trained Gaussian mixture models for dialect and accent recognition. In *INTERSPEECH* (pp. 723–726).

Yu, D., & Deng, L. (2011). Deep learning and its applications to signal and information processing [exploratory DSP]. *IEEE Signal Processing Magazine*, *28*(1), 145–154. http://dx.doi.org/10.1109/MSP.2010.939038.

Zissman, M. (1996). Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *4*(1), 31–44.

Zissman, M. A., & Berkling, K. (2001). Automatic language identification. *Speech Communication*, *35*(1–2), 115–124.